

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-076064

(43)Date of publication of application : 14.03.2000

(51)Int.Cl.

G06F 9/06

(21)Application number : 10-246962

(71)Applicant : HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 01.09.1998

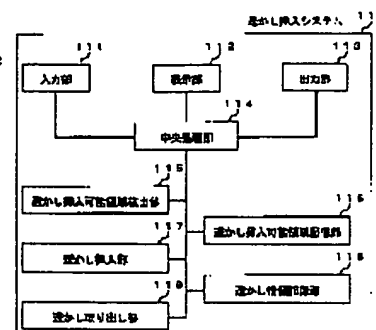
(72)Inventor : TODA JUNICHI

(54) METHOD AND SYSTEM FOR INSERTING WATERMARK INTO PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To easily specify the generation source of illegal use if the illegal use is found by selecting several instruction statements which exert no influence on the program even when rearranged, and replacing the instruction statements at random after inserting a calculation expression regarding a dummy variable to an area between the instruction statements, and saving the order of the rearranged instruction statements as watermark information.

SOLUTION: The creator of the program inputs the program ID of the program, into which a watermark is wanted to insert, and information of users being destinations of the program from an input part 111. A central processing part 114 refers to watermark insertable area information in a watermark insertable area storage part 116 according to the inputted program ID and sends the watermark insertable area information to a watermark information insertion part 117. The watermark insertion part 117 inserts a calculation expression regarding a dummy variable to an area between the instruction statements which can be rearranged according to the watermark insertable area information. After the calculation expression is inserted, the watermark insertion part 117 rearranges the instruction statements at random.



LEGAL STATUS

[Date of request for examination]

26.05.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-76064

(P2000-76064A)

(43) 公開日 平成12年3月14日 (2000.3.14)

(51) Int.Cl.⁷

G 0 6 F 9/06

識別記号

5 5 0

F I

G 0 6 F 9/06

テーマコード(参考)

5 5 0 Z 5 B 0 7 6

審査請求 未請求 請求項の数5 O L (全 8 頁)

(21) 出願番号 特願平10-246962

(22) 出願日 平成10年9月1日 (1998.9.1)

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会
社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 遠田 潤一

神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内

(74) 代理人 100083552

弁理士 秋田 収喜

Fターム(参考) 5B076 FB13

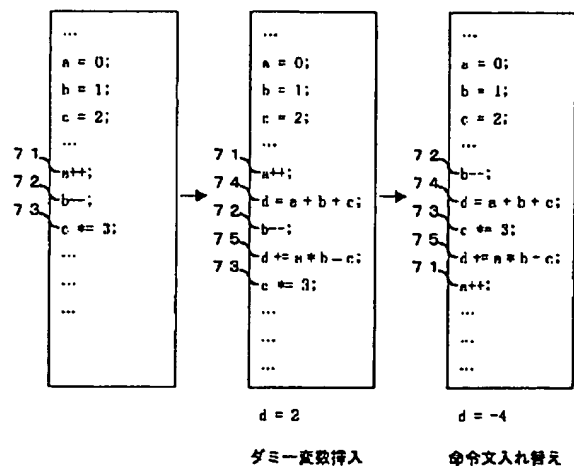
(54) 【発明の名称】 プログラムへの透かしの挿入方法及びシステム

(57) 【要約】

【課題】 コンピュータプログラムの不正利用が発覚した際に発生源を特定できる手段を備えることによって、作成者でない者による不正利用を安価な構成で抑止する。

【解決手段】 プログラムのソースコードから、入れ替えてもプログラムの仕様に影響がない命令文をいくつか取り出し、命令文と命令文との間に、ダミー変数に関する計算式を挿入し、命令文同士をランダムに入れ替え、入れ替えた命令文の順序を透かし情報とする。

図 7



【特許請求の範囲】

【請求項1】 コンピュータを用いて透かしを挿入したいプログラムのソースコードから、順序を入れ替えてもプログラムの仕様に影響がない命令文を複数個取り出し、その命令文と命令文との間に、ダミー変数に関する計算式を挿入し、命令文同士をランダムに入れ替え、入れ替えた命令文の順序を透かし情報として保存することを特徴とするプログラムへの透かしの挿入方法。

【請求項2】 前記透かし情報に対し配付先別の識別情報を付加して記憶装置に保存しておくことを特徴とする請求項1記載のプログラムへの透かしの挿入方法。

【請求項3】 前記命令文を入れ替えたソースコードをコンパイルした際に生成されたアセンブラコードに難読化を施すことを特徴とする請求項1または2記載のプログラムへの透かしの挿入方法。

【請求項4】 プログラムのソースコードを入力する入力手段と、配付先ユーザの識別情報等を表示する表示手段と、透かしの挿入に関する全体の処理を制御する制御手段と、前記入力手段により入力されたプログラムのソースコードから透かしを挿入できる部分を検出する検出手段と、検出した透かしの挿入可能部分を記憶する第1の記憶手段と、検出した透かしの挿入可能部分に透かしを挿入する透かし挿入手段と、挿入した透かしに関する情報を記憶する第2の記憶手段と、透かしを挿入した実行形式プログラムを出力する出力手段と、実行形式プログラムから透かしを取り出す手段とを備えることを特徴とする透かし挿入システム。

【請求項5】 プログラムに透かしを挿入するための透かし挿入用プログラムを記録した記録媒体であって、透かしを挿入したいプログラムのソースコードから、透かしを挿入できる部分を検出する第1の処理と、この第1の処理で検出した部分に透かしを挿入する第2の処理と、挿入した透かしに関する情報を記憶装置に記憶する第3の処理と、透かしを挿入した実行形式プログラムを出力する第4の処理と、実行形式プログラムから透かしを取り出す第5の処理と、を含むコンピュータが実行可能な透かし挿入用プログラムが記録されていることを特徴とする透かし挿入処理用記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、コンピュータプログラムの不正コピー等による不正行為があった場合に、その不正行為の発生源を特定可能にする透かしの挿入方法及びシステムに関するものである。

【0002】

【従来の技術】 情報化の発展、コンピュータの普及により、コンピュータプログラムの重要性は増加している。

コンピュータプログラムの最も特筆すべき特徴は、それ自体がデジタル情報であり、オリジナルと同品質のコピーを高速かつ安価に製造できるという生産性にある。しかしながら、そういった利点がある反面、不正コピー、プログラムコードやアルゴリズムの盗用、改ざんといった不正利用が容易に行われてしまう危険性がある。こういった背景から、作成者でない者によるコンピュータプログラム（ソフトウェア）の不正利用を防止できる環境を作る必要があると考えられる。

【0003】 現在提案されている主なソフトウェア保護手法としては、例えば特開平8-95777号公報「ソフトウェア利用制御装置」に開示されているように、各種のコンテンツをネットワークを通じて有料で流通させる、いわゆる「超流通」などの形態において特別なハードウェア（サーバコンピュータ）を使った方法や、特開平10-27123号公報「コンピュータソフトウェアのコピープロテクト方法」に開示されているように、コピープロテクトをかける方法などが挙げられる。

【0004】

【発明が解決しようとする課題】 しかしながら、「超流通」による方法では、コストが嵩むという問題がある。また、コピープロテクトのように記録媒体を用いる方法では、ネットワークを介してコンピュータプログラムを配布する場合に利用できないという問題がある。作成者にとって安心できる環境を作るためには、低コストで、改ざんや盗用に強い耐タンパな性質をソフトウェア自身に持たせる必要がある。しかし、現実にはコンピュータプログラムはデジタル情報であるため、コピーなどを容易に行うことが可能であり、不正利用自身を防ぐのは難しいと考えられる。

【0005】 本発明の目的は、不正利用自体を防ぐのではなく、不正利用が発覚した際に、その発生源（不正コピー者）を容易に特定できるような仕掛けをコンピュータプログラムに具備させることによって、作成者でない者による不正利用を安価な構成で抑止することができるコンピュータプログラムへの透かし挿入方法及びシステムを提供することにある。

【0006】

【課題を解決するための手段】 前記目的を達成するために、本発明では、現在、画像などの不正利用を防止する技術として使われている電子透かし手法の考え方をコンピュータプログラムに応用する。電子透かし手法とは、例えば特開平9-191394号公報「電子的透かし挿入方法」に示されているように、画像、音声などの著作権を保護する技術の1つであり、画像、音声データの中にID情報を埋め込むことにより画像データの流用、無断転載などの不正利用を防ぐ手法である。

【0007】 前記電子透かし手法の特徴は、

1. IDを埋め込んだことによりデータが劣化しない
2. ユーザが埋め込まれたIDを取り除くとデータが劣

化する（ユーザがIDを取り除けない）という点にある。

【0008】本発明のプログラムへの透かし挿入方法は、透かしの挿入したいプログラムのソースコードから、互いに順序を入れ替えてもプログラムの仕様に影響がない命令文をいくつか選び、その命令文と命令文との間に、ダミー変数に関する計算式を挿入した後、命令文同士をランダムに入れ替え、その入れ替えた命令文の順序を透かし情報として保存することを特徴とする。

【0009】本発明における「透かし」とは、プログラムの仕様に影響を与えず、かつ作成者にとっては意味があるが、利用者側にとっては無意味または存在すること自体が分からないような情報を挿入することである。本発明では、「透かし」の仕方として、ダミー変数を挿入し、そのダミー変数を挿入した命令文同士をランダムに入れ替えるという方法を採用している。

【0010】「透かし情報」とは、上記の方法によって挿入した「透かし」の仕方を表わす情報であり、本発明では、入れ替えた命令文の順序の情報を採用している。

【0011】入替え可能な命令文としては、例えば変数の代入文がある。ダミー変数は、不正なユーザによるプログラムの解析を困難にして、透かしの改ざんを防ぐために用いるものである。

【0012】また、ダミー変数に関する計算式は、入替え可能な命令文で使われている変数を参照することとする。さらに、不正なユーザによるプログラムの解析をより困難にするために、命令文を入れ替えた後ソースコードをコンパイルした際に、生成されたアセンブラコードに対して難読化を施してもよい。難読化とは、例えば電子情報通信学会論文誌V o l . J 8 0 - D - I N o . 7「ループを含むプログラムを難読化する方法の提案」（門田、高田、鳥居）に示されているように、プログラムの仕様を保ちながらその解析を困難にする変換のことである。

【0013】上記の手段によって、不正なユーザが「透かし」を改ざんしようとして命令文を入れ替えると、ダミー変数の計算式が入れ替えた命令文で使われている変数を参照していることにより、ダミー変数の計算結果が変わってしまう。よって、不正なユーザにとっては、「透かし」を改ざんすると、プログラムの仕様が保たれているかどうかのわかりにくくなり、プログラム仕様を犠牲にしてまで改ざんしようとする行為を断念せざるを得なくなり、実質的に不正使用を抑制することが可能になる。

【0014】一方、「透かし」を改ざんし得たとしても、作成者はその改ざん後のプログラムを解析し、作成者側に保存してある透かし情報と照合することによって改ざんされたものであることを直ちに発見することができる。

【0015】また、前記の「透かし情報」に対し配付先

別のユーザIDを付加して保存しておくことにより、不正コピーが多数出回った場合でも、その不正コピーの「透かし情報」を取り出し、その「透かし情報」に対応するユーザIDを調べることにより、不正コピー元を直ちに特定することができる。この結果、作成者でない者によるコンピュータプログラムの不正利用を抑止することが可能になる。

【0016】

【発明の実施の形態】以下、本発明の一実施形態について図面を用いて説明する。

【0017】図1は、本発明を適用した透かし挿入システムの一実施形態を示すシステム構成図である。

【0018】本実施形態の透かし挿入システム110は、入力部111、表示部112、出力部113、中央処理部114、透かし挿入可能領域検出部115、透かし挿入可能領域記憶部116、透かし挿入部117、透かし情報記憶部118、透かし取り出し部119から構成されており、透かし挿入可能領域の検出、透かしの挿入、透かし情報の取り出しなどの処理を行う。

【0019】入力部111は、透かしの挿入したいプログラムのソースコードなどを入力するためのものである。表示部112は、透かし情報などを表示するためのものである。出力部113は、透かしの挿入した実行形式プログラムなどを出力するためのものである。中央処理部114は、透かし挿入可能領域の検出、透かしの挿入、透かしの取り出しなどの処理を制御する。

【0020】透かし挿入可能領域検出部115は、プログラムのソースコードから透かしの挿入できる部分を検出する。透かし挿入可能領域記憶部116は、プログラムのソースコードのうち透かしの挿入できる部分を記憶する。透かし挿入部117は、プログラムのソースコードに透かしの挿入し、ソースコードをコンパイルする。透かし情報記憶部118は、挿入した透かしの情報を記憶する。透かし取り出し部119は、実行形式プログラムから透かしを取り出す。

【0021】本実施形態の透かし挿入システム110は、次に挙げる3つの特徴的な処理を行う。

- 【0022】（1）透かし挿入可能領域の検出
- （2）透かしの挿入
- （3）透かしの取り出し

透かし挿入可能領域の検出処理は、挿入対象のプログラム中に透かしの挿入できる場所を検出するための処理であり、挿入対象のプログラムに透かしを実際に挿入する前に、最初に1回だけ行う。透かしの挿入処理は、配布先のユーザを識別するIDを埋め込むための処理であり、配布対象のプログラムをユーザに配布するたびに作成者の管理化または作成者の委任を受けた管理者の管理下で行う。

【0023】透かしの取り出し処理は、プログラムの配布先のユーザを特定するための処理であり、プログラム

の不正利用を調査するときに行う。プログラムの配布先を特定したい場合の例としては、プログラムの不正コピーが出回っており、不正コピーの出所を突き止めたい場合などがある。以下3つの処理について詳しく説明する。

【0024】まずはじめに、透かし挿入可能領域の検出処理について図2の透かし挿入可能領域検出フロー図を用いて説明する。プログラムの作成者は、透かしを挿入したいプログラムのソースコードを入力部111から入力する（ステップ21）。プログラムのソースコードが入力されると、中央処理部114は、入力されたプログラムに対してプログラムIDを一意に割り当てる（ステップ22）。プログラムIDを割り当てると、中央処理部114は、ソースコードを透かし挿入可能領域検出部115へ送る。透かし挿入可能領域検出部115は、送られたソースコードを解析して、相互に入れ替えてもプログラムの仕様に影響がないような命令文の組を検出する（ステップ23）。

【0025】図3は、ステップ23における透かし挿入可能領域検出処理の説明図である。図3において、命令文31、命令文32および命令文33は、それぞれ異なる変数を参照する代入文である。従って、命令文31、命令文32、命令文33を相互に入れ替えてもプログラムの仕様に影響はない。入替え可能な命令文を検出すると、透かし挿入可能領域検出部115は、命令文の情報を透かし挿入可能領域として中央処理部114へ送る。中央処理部114は、送られた透かし挿入可能領域と、この透かし挿入可能領域を含むプログラムのプログラムIDとを透かし挿入可能領域記憶部116へ送る（ステップ24）。

【0026】図4は、透かし挿入可能領域記憶部116に記憶する透かし挿入可能領域情報の例である。透かし挿入可能領域情報40は、プログラムを識別するためのプログラムID41、プログラムのオリジナルソースコード42、入替え可能な命令文を表す透かし挿入可能領域43から構成されている。

【0027】図5は、透かし挿入可能領域43の例を示すものであり、入替え可能な命令文のソースコード中のソース行番号51、入替え可能な命令文52、命令文を識別するためのコードID53から構成されている。なお、この図5の例は図3の命令文に対応している。なお、図3の例では、透かし挿入可能領域43は1箇所だけであるが、2箇所以上あってもよい。ただし、その場合は、コードID53の表記を工夫して（例、A1、A2、…、B1、B2、…）、どの命令文が相互に入替え可能であるかを識別できるようにしておく必要がある。

【0028】透かし挿入可能領域情報40を送ると、中央処理部114は、表示部112に割り当てたプログラムIDを表示して（ステップ25）処理を終了する。プログラムIDは、透かしの挿入処理の際、透かし挿入シ

ステム110に透かしを挿入したいプログラムを識別させるために、プログラムの作成者が入力する情報である。このため、プログラムの作成者は、表示されたプログラムIDを記録しておく必要がある。

【0029】次に、透かしの挿入処理について図6の透かし挿入フロー図を用いて説明する。プログラムの作成者は、透かしを挿入したいプログラムのプログラムIDと、プログラムの配布先のユーザ情報を入力部111から入力する（ステップ61）。プログラムIDとユーザ情報が入力されると、中央処理部114は、入力されたプログラムIDに基づいて、透かし挿入可能領域記憶部116から透かし挿入可能領域情報40を参照し、その透かし挿入可能領域情報40を透かし情報挿入部117へ送る（ステップ62）。

【0030】透かし挿入可能領域情報40が送られてくると、透かし挿入部117は、前記透かし挿入可能領域情報40に基づいて入れ替え可能な命令文と命令文の間にダミー変数に関する計算式を挿入する（ステップ63）。ダミー変数は、透かしの改ざん防止のためだけに用いるものである。ダミー変数の計算式は、入れ替える命令文で使われている変数を参照することとする。なお、ダミー変数の計算結果は、プログラムの仕様や、透かし情報自体とは何ら関連性がない。計算式を挿入すると、透かし挿入部117は、入替え可能な命令文をランダムに入れ替える（ステップ64）。

【0031】図7は、ステップ63とステップ64における透かし挿入処理の説明図である。命令文71、命令文72および命令文73は透かし挿入可能領域である。ステップ63において、命令文71と命令文72との間に、ダミー変数に関する計算式74が挿入される。また、命令文72と命令文73との間に、ダミー変数に関する計算式75が挿入される。計算式74と計算式75は、命令文71、命令文72および命令文73で使われている変数a、b、およびcを参照している。ステップ64において、命令文71、命令文72および命令文73をランダムに入れ替える。

【0032】命令文を入れ替えると、透かし挿入部117は、ソースコードをコンパイルする（ステップ65）。ソースコードをコンパイルすると、透かし挿入部117は、入替えた命令文の順序情報を透かし情報として中央処理部114へ送る。中央処理部114は、送られた透かし情報と、透かし情報に対応するプログラムIDと、入力されたユーザ情報とを透かし情報記憶部118へ送り、記憶させる（ステップ66）。

【0033】図8は、透かし情報記憶部118に記憶する透かし情報80の例である。透かし情報80は、プログラムを識別するためのプログラムID81、入れ替えた命令文の順序を表す命令文順序情報82、プログラム配布先のユーザを識別するためのユーザID83、および付加的な情報84から構成されている。ユーザID8

3は、透かし情報記憶部118が透かし情報80を記憶させて保存する際、一意に割り当てる。付加的な情報84の例として、配布先のユーザの氏名、住所、電話番号、メールアドレスなどが挙げられる。

【0034】図9は、命令文順序情報82の例である。命令文順序情報82は、命令文のコードID91、命令文に対応するアセンブラコード92、アセンブラコードのアセンブラ行番号93から構成されている。透かし情報80が送られると、透かし情報記憶部118は、透かし情報80におけるプログラムID81と命令文順序情報82が、すでに記憶されている透かし情報80と重複していないかどうかをチェックする(ステップ67)。重複している場合は、ステップ65に戻って命令文を入れ替え直す。重複していない場合は、透かし情報記憶部118は、送られた透かし情報80にユーザID83を付加して透かし情報80を記憶して保存する(ステップ68)。

【0035】この重複チェックの処理により、ユーザID毎に異なる透かしが挿入され、その透かし情報が記憶部118に保存されることになる。

【0036】透かし情報80が記憶されると、中央処理部114は、出力部113から透かしを挿入した実行形式プログラムを出力して(ステップ69)処理を終了する。このようにして透かしを挿入した実行形式プログラムはユーザに配布される。

【0037】次に、ユーザに配付した実行形式プログラムから透かしを取り出す処理について図10の透かし取り出しフロー図を用いて説明する。プログラムの作成者は、透かしを取り出したいプログラムの実行形式ファイルを入力部111から入力する(ステップ101)。さらに、プログラムIDを入力部111から入力する(ステップ102)。

【0038】実行形式プログラムとプログラムIDが入力されると、中央処理部114は、入力されたプログラムIDに基づいて、透かし情報記憶部118に保存しておいた当該プログラムIDに対応する透かし情報80を取り出し、前記実行形式プログラムと共に透かし取り出し部119へ送る(ステップ103)。この場合、同じプログラムIDのプログラムを複数のユーザに配付した場合には、そのプログラムIDを持つ全ての透かし情報80が取り出されて透かし取り出し部119へ送られる。

【0039】実行形式プログラムと透かし情報80が送られてくると、透かし取り出し部119は、実行形式プログラムを逆アセンブルした後、透かしに用いた命令文の順序を解析し、命令文順序情報82を生成する。この場合、透かしに用いた命令文の範囲は、透かし情報記憶部118から取り出した透かし情報80内の命令文順序情報82におけるアセンブラ行番号93の範囲によって分かる。

【0040】命令文順序情報82を生成したならば、透かし取り出し部119は、生成した命令文順序情報82と透かし情報記憶部118から取り出した透かし情報80内の命令文順序情報82とを照合し、両者が一致する(または同じ意味を持つ)命令文順序情報82を持つ透かし情報80を選択抽出し、その選択抽出した透かし情報80を中央処理部114へ送る(ステップ104)。

【0041】図11は、ステップ103とステップ104における透かし取り出し処理の説明図である。実行形式プログラムが逆アセンブルされると、命令文の順序が解析されて、命令文順序情報82が生成される。そして、生成された命令文順序情報82と透かし情報記憶部118から取り出した透かし情報80内の命令文順序情報82とが照合され、両者が一致する(または同じ意味を持つ)命令文順序情報82を持つ透かし情報80が選択抽出され、その選択抽出された透かし情報80内のユーザID83や付加的な情報84が取り出される。

【0042】中央処理部114は、取り出されたユーザID83や付加的な情報84を表示部112に表示して(ステップ105)処理を終了する。

【0043】このように本実施形態では、命令文の順序を透かしとしてプログラムに挿入している。従って、不正なユーザが、プログラムの仕様を変えずに透かしを改ざんするためには、透かし挿入可能領域の命令文を入れ替える必要がある。ところが、透かし挿入可能領域の命令文を入れ替えると、ダミー変数の計算結果が変わってしまう。図7の例では、命令文入れ替え前はダミー変数dの計算結果は「2」であるが、命令文入れ替え後はダミー変数dの計算結果は「-4」となる。このため、不正なユーザが透かし挿入可能領域の命令文を入れ替えたとしても、不正なユーザにとっては命令文入れ替え後もプログラムの仕様が保たれているかどうかのわかりにくくなり、プログラム仕様を犠牲にしてまで改ざんしようとする行為を断念せざるを得なくなり、実質的に不正使用を抑制することが可能になる。

【0044】一方、「透かし」を改ざんし得たとしても、作成者はその改ざん後のプログラムを解析し、作成者側に保存してある透かし情報と照合することによって改ざんされたものであることを直ちに発見することができる。また、「透かし情報」に対し配付先別のユーザIDを付加して保存しておくことにより、不正コピーが多数出回った場合でも、その不正コピーの「透かし」を取り出し、その「透かし」に対応するユーザIDを調べることにより、不正コピー元を直ちに特定することができる。この結果、作成者でない者によるコンピュータプログラムの不正利用を抑止することが可能になる。また、透かし挿入後に難読化を施せば、不正なユーザがプログラムを解析することがより一層困難になり、不正利用を抑制する上でさらに効果的である。なお、上記実施形態において、透かし挿入システム110における透かし挿

入可能領域検出機能、透かし挿入可能領域記憶機能、透かし挿入機能、透かし情報記憶機能、透かし取り出し機能は、中央処理部114で実行可能なコンピュータプログラムで構成されるものであるが、これらの機能を構成するプログラムの全部または一部をCD-ROM等の記憶媒体に記録し、あるいはインターネット等の通信媒体を介して有料でプログラム作成者に提供することができる。上記機能をプログラム作成者のコンピュータで実行することにより、それぞれのプログラム作成者が作成したコンピュータプログラムに対し独自の「透かし」を挿入し、不正利用を抑止し、著作権の保護に貢献することができる。

【0045】

【発明の効果】以上説明したように、本発明の透かし挿入方法を用いれば、プログラム仕様を犠牲にしてまで改ざんしようとする行為を断念せざるを得なくなり、不正使用を安価な構成で抑制することが可能になる。また、「透かし」を改ざんし得たとしても、作成者はその改ざん後のプログラムを解析し、作成者側に保存してある「透かし情報」と照合することによって改ざんされたものであることを直ちに発見することができる。さらに、「透かし情報」に対し配付先別のユーザIDを付加して保存しておくことにより、不正コピーが多数出回った場合でも、その不正コピーの「透かし」を取り出し、その「透かし」に対応するユーザIDを調べることにより、不正コピー元を直ちに特定することができる。この結果、作成者でない者によるコンピュータプログラムの不正利用を抑止することが可能になる。また、透かし挿入後に難読化を施せば、不正なユーザがプログラムを解析することがより一層困難になり、不正利用を抑制する上でさらに効果的である。

【図面の簡単な説明】

【図1】本発明の実施形態を示すシステム構成図である。

【図2】図1の実施形態における透かし挿入可能領域検出フロー図である。

【図3】図1の実施形態における透かし挿入可能領域検出処理の説明図である。

【図4】図1の実施形態における透かし挿入可能領域情報を示す図である。

【図5】図1の実施形態における透かし挿入可能領域を示す図である。

【図6】図1の実施形態における透かし挿入フロー図である。

【図7】図1の実施形態における透かし挿入処理の説明図である。

【図8】図1の実施形態における透かし情報を示す図である。

【図9】図1の実施形態における命令文順序を示す図である。

【図10】図1の実施形態における透かし取り出しフロー図である。

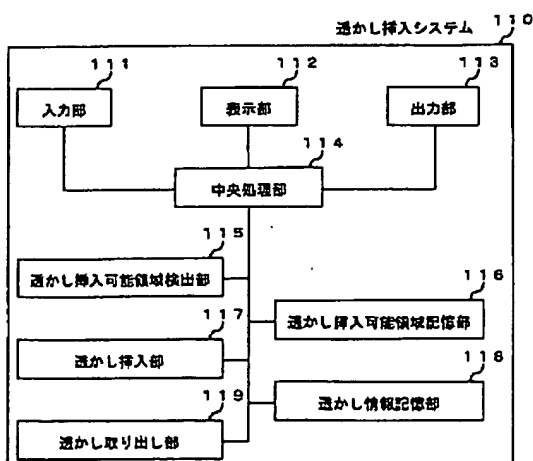
【図11】図1の実施形態における透かし取り出し処理の説明図である。

【符号の説明】

80…透かし情報、82…命令文順序情報、83…ユーザID、110…透かし挿入システム、111…入力部、112…表示部、113…出力部、114…中央処理部、115…透かし挿入可能領域検出部、116…透かし挿入可能領域記憶部、117…透かし挿入部、118…透かし情報記憶部、119…透かし取り出し部。

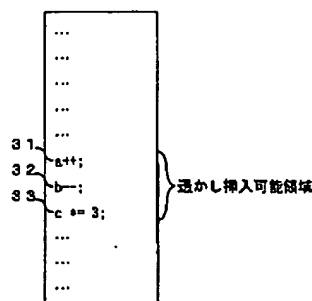
【図1】

図1



【図3】

図3



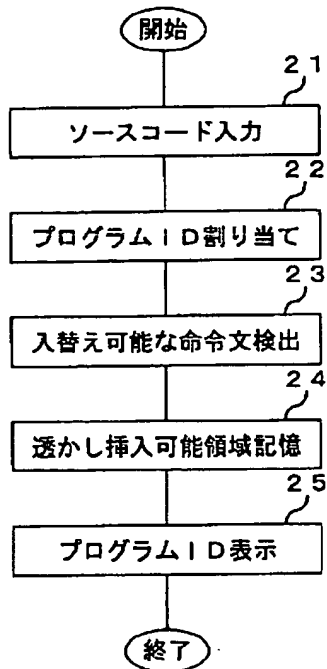
【図5】

図5

51 ソース行番号	52 命令文	53 コードID
128	a++;	1
129	b--;	2
130	c*=3;	3

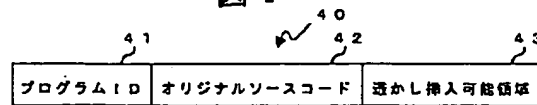
【図2】

図 2



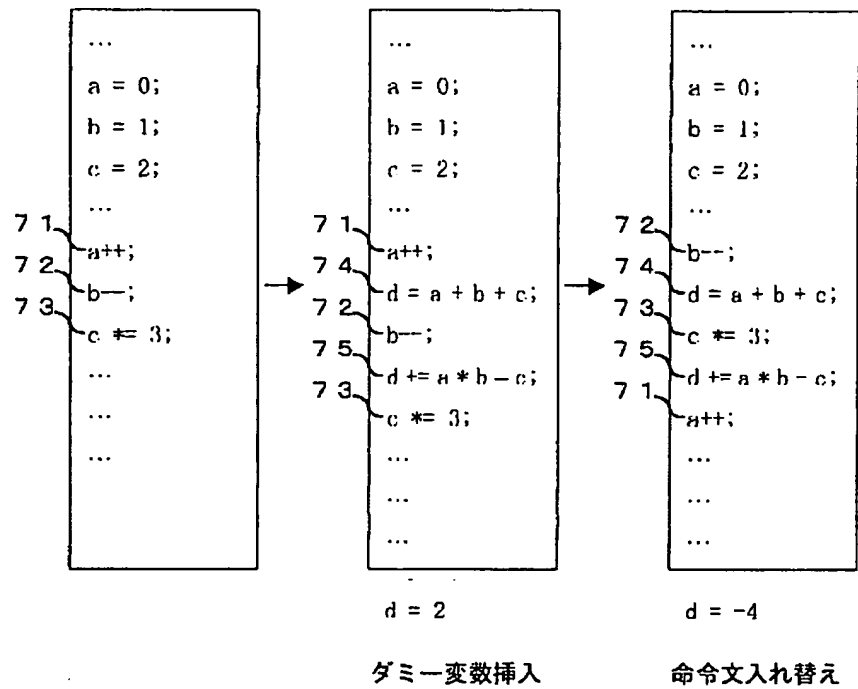
【図4】

図 4



【図7】

図 7



【図8】

図 8

8 1	8 2	8 0	8 3	8 4
プログラムID	命令文順序情報	ユーザID	...	

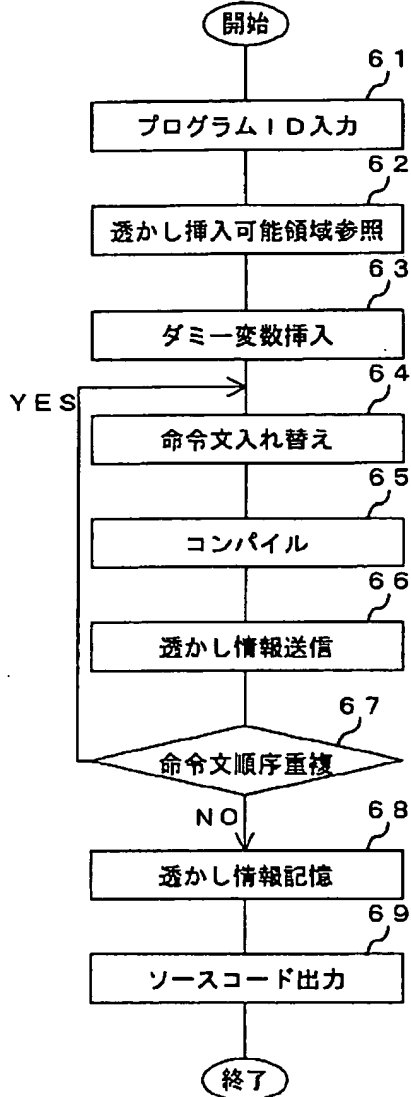
【図9】

図 9

9 1	9 2	9 3
コードID	アセンブラ命令	アセンブラ行番号
2	DEC DX	8 1 3
3	MOV CX, 3	8 1 8
	MUL CX	8 1 9
1	INC BX	8 2 6

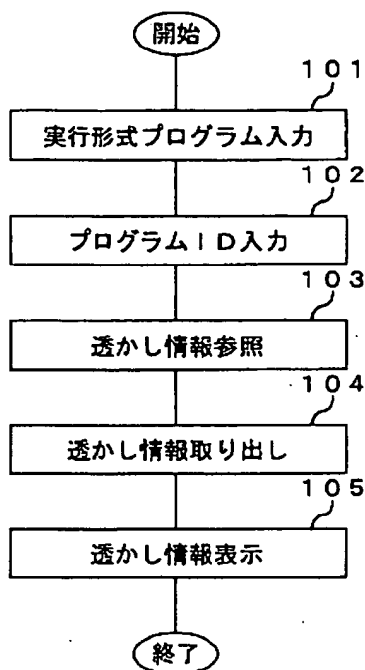
【図6】

図 6



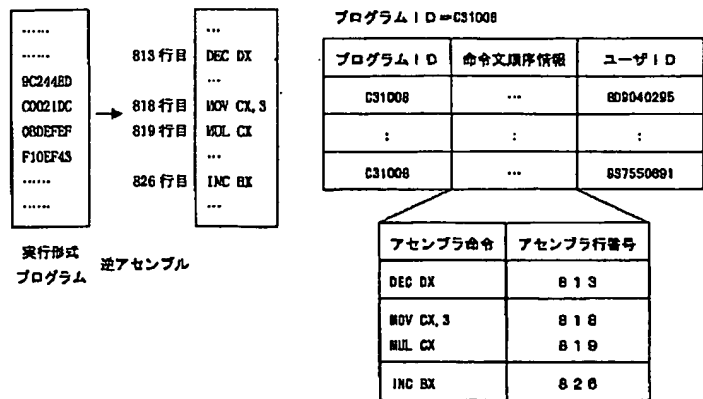
【図10】

図 10



【図11】

図 11



ユーザID=857550891